# Clickhouse服务手册

目录

[Clickhouse服务手册 1](#_Toc520452910)

[安装部署 2](#_Toc520452911)

[系统环境 2](#_Toc520452912)

[安装 2](#_Toc520452913)

[启动 2](#_Toc520452914)

[配置项 3](#_Toc520452915)

[搭建数据库 5](#_Toc520452916)

[数据格式 5](#_Toc520452917)

[引擎 6](#_Toc520452918)

[建表语句 7](#_Toc520452919)

[更改语句 7](#_Toc520452920)

[ptd\_data建表语句： 7](#_Toc520452921)

[分布式搭建 9](#_Toc520452922)

[zookeeper高复用 9](#_Toc520452923)

[hangout插件 10](#_Toc520452924)

[查询数据 11](#_Toc520452925)

[查询语法 11](#_Toc520452926)

[查询函数 12](#_Toc520452927)

[性能分析 12](#_Toc520452928)

[现有问题 12](#_Toc520452929)

[消费kafka数据 12](#_Toc520452930)

[hangout插件 12](#_Toc520452931)

[复杂JSON数据 12](#_Toc520452932)

[动态修改表 13](#_Toc520452933)

[数据备份 13](#_Toc520452934)

[事务支持 13](#_Toc520452935)

## 安装部署

### 系统环境

CentOS-7-x86\_64-DVD-1708

clickhouse 1.1.54388

### 安装

安装curl：

sudo yum install –y curl

更新依赖：

curl -s https://packagecloud.io/install/repositories/altinity/clickhouse/script.rpm.sh | sudo bash

检查clickhouse包：

sudo yum list 'clickhouse\*'

安装clickhouse：

sudo yum install -y 'clickhouse\*'

Q&A：

q：依赖无法更新或使用

a：从<https://packagecloud.io/altinity/clickhouse>下载rpm包，手动安装

### 启动

启动服务：

sudo service clickhouse-server start

启动打印输出的服务：

clickhouse-server --config-file=/etc/clickhouse-server/config.xml

停止服务：

sudo service clickhouse-server stop

日志信息：

存在 /var/log/clickhouse-server/

启动客户端：

clickhouse-client（默认数据库，默认用户，默认localhost:9000）

clickhouse-client –h YourHost –u YourUserName --password password

### 配置项

配置文件：

全局配置 /etc/clickhouse-server/config.xml

用户配置 /etc/clickhouse-server/user.xml

配置监听：

在config.xml修改配置<listen\_host>0.0.0.0</listen\_host>

配置用户：

在user.xml修改配置

添加用户样例

<yourname>

<password>password</password>

<networks incl="networks" replace="replace">

<ip>::/0</ip>

</networks>

<profile>default</profile>

<quota>default</quota>

</yourname>

配置分布式服务：

修改config.xml

配置样例

<remote\_servers>

<distribute>

<shard>

<!-- Optional. Shard weight when writing data. Default: 1. -->

<weight>1</weight>

<!-- Optional. Whether to write data to just one of the replicas. Default: false (write data to all replicas). -->

<internal\_replication>false</internal\_replication>

<replica>

<host>192.168.18.143</host>

<port>9000</port>

</replica>

</shard>

<shard>

<weight>1</weight>

<internal\_replication>false</internal\_replication>

<replica>

<host>192.168.18.142</host>

<port>9000</port>

</replica>

</shard> </distribute>

</remote\_servers>

配置Zookeeper高复用：

修改config.xml

<zookeeper>

<node index="1">

<host>host1</host>

<port>2181</port>

</node>

<node index="2">

<host>host2</host>

<port>2181</port>

</node>

</zookeeper>

<macros>

<replica>LocalHostName</replica>

</macros>

## 搭建数据库

### 数据格式

输入和输出格式：

format in out

|  |  |  |  |
| --- | --- | --- | --- |
| [TabSeparated](https://clickhouse.yandex/docs/en/interfaces/formats/#tabseparated) |  | ✔ | ✔ |
| [TabSeparatedRaw](https://clickhouse.yandex/docs/en/interfaces/formats/#tabseparatedraw) |  | ✗ | ✔ |
| [TabSeparatedWithNames](https://clickhouse.yandex/docs/en/interfaces/formats/#tabseparatedwithnames) |  | ✔ | ✔ |
| [TabSeparatedWithNamesAndTypes](https://clickhouse.yandex/docs/en/interfaces/formats/#tabseparatedwithnamesandtypes) |  | ✔ | ✔ |
| [CSV](https://clickhouse.yandex/docs/en/interfaces/formats/#csv) |  | ✔ | ✔ |
| [CSVWithNames](https://clickhouse.yandex/docs/en/interfaces/formats/#csvwithnames) |  | ✔ | ✔ |
| [Values](https://clickhouse.yandex/docs/en/interfaces/formats/#values) |  | ✔ | ✔ |
| [Vertical](https://clickhouse.yandex/docs/en/interfaces/formats/#vertical) |  | ✗ | ✔ |
| [VerticalRaw](https://clickhouse.yandex/docs/en/interfaces/formats/#verticalraw) |  | ✗ | ✔ |
| [JSON](https://clickhouse.yandex/docs/en/interfaces/formats/#json) |  | ✗ | ✔ |
| [JSONCompact](https://clickhouse.yandex/docs/en/interfaces/formats/#jsoncompact) |  | ✗ | ✔ |
| [JSONEachRow](https://clickhouse.yandex/docs/en/interfaces/formats/#jsoneachrow) |  | ✔ | ✔ |
| [TSKV](https://clickhouse.yandex/docs/en/interfaces/formats/#tskv) |  | ✔ | ✔ |
| [Pretty](https://clickhouse.yandex/docs/en/interfaces/formats/#pretty) |  | ✗ | ✔ |
| [PrettyCompact](https://clickhouse.yandex/docs/en/interfaces/formats/#prettycompact) |  | ✗ | ✔ |
| [PrettyCompactMonoBlock](https://clickhouse.yandex/docs/en/interfaces/formats/#prettycompactmonoblock) |  | ✗ | ✔ |
| [PrettyNoEscapes](https://clickhouse.yandex/docs/en/interfaces/formats/#prettynoescapes) |  | ✗ | ✔ |
| [PrettySpace](https://clickhouse.yandex/docs/en/interfaces/formats/#prettyspace) |  | ✗ | ✔ |
| [RowBinary](https://clickhouse.yandex/docs/en/interfaces/formats/#rowbinary) |  | ✔ | ✔ |
| [Native](https://clickhouse.yandex/docs/en/interfaces/formats/#native) |  | ✔ | ✔ |
| [Null](https://clickhouse.yandex/docs/en/interfaces/formats/#null) |  | ✗ | ✔ |
| [XML](https://clickhouse.yandex/docs/en/interfaces/formats/#xml) |  | ✗ | ✔ |
| [CapnProto](https://clickhouse.yandex/docs/en/interfaces/formats/#capnproto) |  | ✔ | ✔ |

表内数据格式：

* [UInt8, UInt16, UInt32, UInt64, Int8, Int16, Int32, Int64](https://clickhouse.yandex/docs/en/data_types/int_uint/)
* [Float32, Float64](https://clickhouse.yandex/docs/en/data_types/float/)
* [Boolean values](https://clickhouse.yandex/docs/en/data_types/boolean/)
* [String](https://clickhouse.yandex/docs/en/data_types/string/)
* [FixedString(N)](https://clickhouse.yandex/docs/en/data_types/fixedstring/)
* [Date](https://clickhouse.yandex/docs/en/data_types/date/)
* [DateTime](https://clickhouse.yandex/docs/en/data_types/datetime/)
* [Enum](https://clickhouse.yandex/docs/en/data_types/enum/)
* [Array(T)](https://clickhouse.yandex/docs/en/data_types/array/)
* [AggregateFunction(name, types\_of\_arguments...)](https://clickhouse.yandex/docs/en/data_types/nested_data_structures/aggregatefunction/)
* [Tuple(T1, T2, ...)](https://clickhouse.yandex/docs/en/data_types/tuple/)
* Nested data structures
* Special data types

Q&A：

q:插入空数据时报错，如(String) value=null

a:所有数据都没有null值，Int型的null会自动转换为0，String、Array会报错

### 引擎

MergeTree

* + - [MergeTree](https://clickhouse.yandex/docs/en/operations/table_engines/mergetree/)
    - [ReplacingMergeTree](https://clickhouse.yandex/docs/en/operations/table_engines/replacingmergetree/)
    - [SummingMergeTree](https://clickhouse.yandex/docs/en/operations/table_engines/summingmergetree/)
    - [AggregatingMergeTree](https://clickhouse.yandex/docs/en/operations/table_engines/aggregatingmergetree/)
    - [CollapsingMergeTree](https://clickhouse.yandex/docs/en/operations/table_engines/collapsingmergetree/)
    - [GraphiteMergeTree](https://clickhouse.yandex/docs/en/operations/table_engines/graphitemergetree/)
    - [TinyLog](https://clickhouse.yandex/docs/en/operations/table_engines/tinylog/)
    - [Log](https://clickhouse.yandex/docs/en/operations/table_engines/log/)
    - [Memory](https://clickhouse.yandex/docs/en/operations/table_engines/memory/)
    - [Buffer](https://clickhouse.yandex/docs/en/operations/table_engines/buffer/)
    - [External data](https://clickhouse.yandex/docs/en/operations/table_engines/external_data/)

Special

* + - [Distributed](https://clickhouse.yandex/docs/en/operations/table_engines/distributed/)
    - [Dictionary](https://clickhouse.yandex/docs/en/operations/table_engines/dictionary/)
    - [Merge](https://clickhouse.yandex/docs/en/operations/table_engines/merge/)
    - [File](https://clickhouse.yandex/docs/en/operations/table_engines/file/)
    - [Null](https://clickhouse.yandex/docs/en/operations/table_engines/null/)
    - [Set](https://clickhouse.yandex/docs/en/operations/table_engines/set/)
    - [Join](https://clickhouse.yandex/docs/en/operations/table_engines/join/)
    - [View](https://clickhouse.yandex/docs/en/operations/table_engines/view/)
    - [MaterializedView](https://clickhouse.yandex/docs/en/operations/table_engines/materializedview/)

Integrations

* + - [Kafka](https://clickhouse.yandex/docs/en/operations/table_engines/kafka/)
    - [MySQL](https://clickhouse.yandex/docs/en/operations/table_engines/mysql/)

### 建表语句

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] [db.]name [ON CLUSTER cluster]

(

name1 [type1] [DEFAULT|MATERIALIZED|ALIAS expr1],

name2 [type2] [DEFAULT|MATERIALIZED|ALIAS expr2],

...

) ENGINE = engine

### 更改语句

ALTER TABLE [db].name [ON CLUSTER cluster] ADD|DROP|MODIFY COLUMN...

ADD COLUMN name [type] [default\_expr] [AFTER name\_after]

DROP COLUMN name

MODIFY COLUMN name [type] [default\_expr]

### ptd\_data建表语句：

create table test2(a\_vector UInt8,ack Array(UInt32),alert UInt8,bytes UInt32,c2s\_bytes UInt32,c2s\_pkts UInt8,cp Array(String),d\_domain Array(String),d\_domains Array(String),d\_file Array(String),d\_files Array(String),d\_ip Array(String),d\_sengine Array(String),d\_url Array(String),d\_urls Array(String),dev String,domain Array(String),domains Array(String),dst\_city String,dst\_country String,dst\_ip String,dst\_ipv6 String,dst\_isp String,dst\_location\_lat String,dst\_location\_lon String,dst\_mac String,dst\_port UInt32,dst\_region String,dst\_unit String,file Array(String),files Array(String),flow\_id UInt64,flow\_pkg\_id UInt32,has\_pta UInt8,http\_request\_method String,http\_request\_header\_other Array(String),http\_request\_header\_host String,http\_request\_header\_uri String,http\_request\_body\_bytes UInt32,http\_request\_body\_fn String,http\_request\_body\_ft String,http\_request\_body\_direction String,http\_request\_body\_md5 String,http\_response\_header\_other Array(String),http\_response\_body\_bytes UInt32,http\_response\_body\_fn String,http\_response\_body\_ft String,http\_response\_body\_direction String,http\_response\_body\_md5 String,http\_response\_status UInt8,http\_i UInt8,id UInt64,ip Array(String),is\_malicious UInt8,label Array(String),malname Array(String),msg\_download\_bytes UInt32,msg\_download\_fn String,msg\_download\_ft String,msg\_download\_direction String,msg\_download\_md5 String,msg\_upload\_bytes UInt32,msg\_upload\_fn String,msg\_upload\_ft String,msg\_upload\_direction String,msg\_upload\_md5 String,pid String,pkts UInt32,proto Array(String),protoid Array(UInt16),mail Array(String),email\_msg\_cc Array(String),email\_msg\_sub String,email\_msg\_bcc Array(String),email\_msg\_sender String,email\_msg\_from String,email\_msg\_attach Array(String),email\_msg\_to Array(String),email\_msg\_content Array(String),email\_encrypted UInt8,email\_passwd String,email\_clientver String,email\_serverver String,email\_user String,ftp\_download\_bytes UInt32,ftp\_download\_fn String,ftp\_download\_ft String,ftp\_download\_direction String,ftp\_download\_md5 String,ftp\_upload\_bytes UInt32,ftp\_upload\_fn String,ftp\_upload\_ft String,ftp\_upload\_direction String,ftp\_upload\_md5 String,pta\_analysis Array(String),dns\_msg\_record Array(String),dns\_msg\_msg\_type String,users\_dst\_user\_deployment\_location String,users\_dst\_user\_asset\_class String,users\_dst\_user\_id\_card String,users\_dst\_user\_name String,users\_dst\_user\_phonenumber String,users\_dst\_user\_addr String,users\_dst\_user\_department String,users\_dst\_user\_mac String,users\_dst\_user\_username String,users\_dst\_user Array(String),users\_src\_user Array(String),users\_src\_user\_deployment\_location String,users\_src\_user\_asset\_class String,users\_src\_user\_id\_card String,users\_src\_user\_name String,users\_src\_user\_phonenumber String,users\_src\_user\_addr String,users\_src\_user\_department String,users\_src\_user\_mac String,users\_src\_user\_username String,pver String,r\_ip Array(String),risk\_level UInt8,s2c\_bytes UInt32,s2c\_pkts UInt8,seq Array(UInt32),src\_city String,src\_country String,src\_ip String,src\_ipv6 String,src\_isp String,src\_location\_lat String,src\_location\_lon String,src\_mac String,src\_port UInt32,src\_region String,src\_unit String,threat\_level UInt8,day Date,hour DateTime,min DateTime,ts\_end DateTime,ts\_start DateTime,type String,url Array(String),urls Array(String),ver String,ProtoFlowStats\_HTTP\_bytes UInt32,ProtoFlowStats\_HTTP\_pkts UInt32,\_log\_src String,mysql\_encrypted UInt8,mysql\_passwd String,mysql\_rcode UInt8,mysql\_cmd String,mysql\_login UInt8,mysql\_user String,mysql\_sql String)ENGINE=ReplicatedMergeTree('/clickhouse/tables/test2','{replica}',day,(day,hour,min,id),8192);

### 分布式搭建

修改配置文件，见安装部署-配置项

创建Distributed表：

CREATE TABLE test\_all AS test1 ENGINE = Distributed(distribute, default, test1,rand())

在创建test\_all表前，需要先在配置节点的服务器创建test1表

distributed表不存数据，实际数据存在test1表

查询时使用test\_all表

rand()为写入策略，代表随机写入任意可用节点

### zookeeper高复用

修改配置文件，见安装部署-配置项

创建高复用表：

ReplicatedMergeTree('/clickhouse/tables/test2','{replica}',day,(day,hour,min,id),8192)

其中'/clickhouse/tables/test1'为Zookeeper中的路径

day必须为Date格式

（day,hour,min,id）为索引

8192为索引粒度

### hangout插件

官方的kafka engine不支持输入具有复杂结构的json数据，故需要自己实现。

第三方插件hangout是由java编写的从kafka写数据到clickhouse的插件，但依然不支持复杂json数据。不过该插件提供了良好的编程接口，开发人员可以在此基础上处理输入的json数据，处理好之后再输出到clickhouse中。

下载地址：<https://git.antiy.org.cn/huangboqi/hadoop-test>

配置文件：hangout-master/conf/app.xml

配置样例：

inputs:

- NewKafka:

topic:

topicName: 2

codec: json

consumer\_settings:

bootstrap.servers: hostName:9092

value.deserializer: org.apache.kafka.common.serialization.StringDeserializer

key.deserializer: org.apache.kafka.common.serialization.StringDeserializer

group.id: hangout

outputs:

- com.sina.bip.hangout.outputs.Clickhouse:

host: hostName:8123

username: username

password: password

database: default

format: JSONEachRow

table: tableName

bulk\_size: 5

convertNull: ['d\_','r\_','http\_','label','dns\_','users\_','email\_']

具体的实现代码在： /hangout-master/hangout-output-plugins/hangout-output-clickhouse-master/src/main/java/com/sina/bip/hangout/outputs

实现的代码文件为：

JSONEachRow.java

RecordParser.java

## 查询数据

### 查询语法

SELECT [DISTINCT] expr\_list

[FROM [db.]table | (subquery) | table\_function] [FINAL]

[SAMPLE sample\_coeff]

[ARRAY JOIN ...]

[GLOBAL] ANY|ALL INNER|LEFT JOIN (subquery)|table USING columns\_list

[PREWHERE expr]

[WHERE expr]

[GROUP BY expr\_list] [WITH TOTALS]

[HAVING expr]

[ORDER BY expr\_list]

[LIMIT [n, ]m]

[UNION ALL ...]

[INTO OUTFILE filename]

[FORMAT format]

[LIMIT n BY columns]

需要注意的是DISTINCT,GROUP BY,ORDER BY,IN,JOIN操作要缓存

大量的数据，有可能占用服务器大量内存；需要恰当的设置max\_memory\_usage, max\_rows\_to\_group\_by, max\_rows\_to\_sort, max\_rows\_in\_distinct, max\_bytes\_in\_distinct, max\_rows\_in\_set, max\_bytes\_in\_set, max\_rows\_in\_join, max\_ ytes\_in\_join, max\_bytes\_before\_external\_sort, max\_bytes\_before\_external\_group\_by

参考https://clickhouse.yandex/docs/en/operations/settings/settings/

### 查询函数

参考<https://clickhouse.yandex/docs/en/query_language/functions/>

ps：

判断Array(String)格式的字段是否为空用 notempty(Array)

判断Array(String)格式的字段是否含有某些数据用 hasany(Array,[String])

选取含有某些数据的数组用 select arrayfilter(x -> x like(‘%String%’),Array)

## 性能分析

## 现有问题

### 消费kafka数据

官方的clickhouse不支持从kafka输入复杂的json数据，需要自己实现，详见搭建数据库-hangout插件。

### hangout插件

现在数据处理的实现方式需要修改底层代码，不利于迭代升级和上层数据修改。（可能的解决方案：用修改配置的形式来应对不同的数据输入）

额外的中间件和编写的数据处理代码可能会拖慢数据输入的速度。（增加kafka的分区数，代码调优）

### 复杂JSON数据

对于形如”data”:{“a”:1,”b”:”someString”}的JSON字段，将转化为对应的a，和b的格式，如”data\_a”:1,”data\_b”:”someString”。对于更复杂的嵌套字段，将以同样的方式解析。

对于形如”data”:[{“a”:1,”b”:2},{“c”:3,”d”:4}]的JSON字段，将转化为Array(String)格式，如”data”:[“{“a”:1,”b”:2}”,”{“c”:3,”d”:4}”]

### 动态修改表

当上层数据发生变动时，如添加一个字段，表无法动态修改，需要手动用ALTER语句修改，而且如果Array(String)格式的数据，需要同时修改hangout插件。（可能的解决方案：在hangout插件中添加动态修改的模块）

### 数据备份

使用分布式搭建，即Distributed引擎可以设置1~n份冗余数据，尚未测试如何恢复数据。

ReplicatedMergeTree引擎可以通过Zookeeper进行数据同步备份，这种备份方式独立于Distributed引擎的备份方式。

### 事务支持

Clickhouse不支持事务(Transaction),当消费每块kafka数据时，如果发生错误，该块数据将会丢失（会记录到错误日志）。